Cyclometer

Introductory Description:

Mountain Biking is a great outdoor activity which is becoming more popular every day. I enjoy riding my bike, and I wanted to incorporate my senior project with something that pertained to me personally. I asked myself, how can I better enhance my bike riding experience? When I ride my bike I am sometimes curious about how far I have traveled, and how fast I am going. Who wants to keep looking at their watch when they ride? And trying to gauge your distance traveled using landmarks can be difficult. I decided to build a device similar to that on an exercise bike, which keeps track of your speed, distance, and time traveled on a display right in front of you. From research on this subject I found that there are products like these on the market called Cyclometers. My goal is to build a Cyclometer with similar features of those on the market, but with my own style of displaying options, at a competitive price. My device will display the current ride-time at all times, and give the rider the option of what he or she wants to see on the other display line. The rider can either view the current speed, or the total distance traveled.

General Description:

My cyclometer will be a low power, battery operated device. Riders will be able to read their total ride time, current speed, and overall distance traveled from an LCD display. I noticed many cyclometers on the market used very small LCD displays which seemed to me like they would be difficult to read, especially while glancing down at high speeds. My cyclometer will incorporate a larger LCD display for easier viewing.
Designed for use outdoors, my power supply will consist of four AA alkaline batteries. The enclosure will be water resistant and shock resistant to stand up to harsh riding conditions in the Pacific Northwest. The final assembled hardware, pictured in Figure 1.1, will consist of a Liquid Crystal Display, and a four button keypad. The user interface for my cyclometer is quite simple. The design will be simple to use and easy to read. Final Product dimensions are not to exceed 4 inches tall, 4 inches wide and 1 and a half inches thick.

Figure 1.1 Sketch of Assembled Hardware.
**Functional Description of Hardware**

A detailed Cyclometer hardware block diagram is shown in Figure 1.2. The primary hardware parts include the Hall-effect sensor IC, M68HC912B32 micro-controller, Liquid Crystal Display (LCD), 4-Pushbutton Interface, and a Portable Power Supply. This project is software intensive, thus the hardware list is short, and less complex. Figure 1.2 shows a diagram of the Functional Hardware design.

**Figure 1.2 Functional Hardware Design**
**Hall-effect IC:**

The Hall-effect magnetic sensor is really the key component to my cyclometer. The Hall-effect chip is a semiconductor integrated circuit utilizing the Hall-effect. It is designed to operate in the unidirectional magnetic field, especially at low supply voltage. It operates under a supply voltage range of 3.6 volts to 16 volts. Hall effect IC’s are usually used as speed sensors, position sensors, rotation sensors, or small switches. In this case it will be used to detect speed and distance. The Hall-effect will keep track of each completed revolution of my mountain bike’s wheel. Each time the small magnet passes the fixed sensor a small pulse will be sent to the HC12 microcontroller which will count the pulses to be used in algorithms to find speed and total distance traveled. The Hall-effect sensor is perfect for this application because it has a great operating temperate range for outdoors (-40 degrees C to +85 degrees C). It is a non-contact switch needed for the moving part of the wheel, and it will function well during the vibration of a bike ride.

**M68HC912B32:**

The microcontroller in my cyclometer will be performing the task of handling user interface between the push-buttons and the liquid crystal display screen. Also it will be receiving signals from the hall-effect sensor and using written code with conversion algorithms to obtain speed, distance, and time to be displayed to the user. There are smaller microcontrollers on the market and faster, more complex microcontrollers if you have the money, but the HC12 has all the features I need for this project at a decent price. The HC12 has somewhat low power consumption at 45mA, which 4AA batteries can easily power, and my familiarity with the HC12
will speed the development of my code and quicken the pace of hardware connections. Also I already have a 400 plus page manual devoted to the HC12 to help me with questions and guide my design process. Since my project is portable, I must overcome the hurdle of designing my own circuit board with the appropriate components like the HC12, 16 MHz crystal, LCD, and pushbutton and sensor connections. The 32k bytes of Flash memory allotted for the HC12 should be large enough for my code, and the 1k byte of Ram will be used for storing temporary variables. The PDLC and Port A will interface with the LCD, and Port B will interface with the push buttons. Port T will handle the input capture of signals received from the Hall-effect sensor. Power supplied to my cyclometer will consist of four AA batteries wired in series. The 4 AA batteries generate about 6.0 volts which is perfect for powering my HC12, LCD, and Hall-effect IC.

Software Description

The software components for my cyclometer will be written in the C programming language. The C language was chosen because I have some experience with C++ from computer science 140. In addition I gained some exposure to C from the Embedded Systems class I took last year. I felt that some algorithms for this project would be too difficult to write in assembly code, thus C seemed to be the best choice. The following is a list of the modules to be used with a brief description of each.

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCD</td>
<td>This module will handle all display tasks, like initialization, cursor movement, cursor options, prompting user, and displaying information.</td>
</tr>
</tbody>
</table>
OC_DELAY  This mode will control time-slicing, task switching, and prioritizing of events.

KEY  This module will handle functional state transitions and monitor button de-bouncing.

MSDELAY  Module containing 1millisecond delays to aid me in creating my ride-time clock and help me set up my time-slice scheduler.

CYCLOMETER  The main module, consisting of all algorithms to convert data to speed, distance, and calories burned. Also handles user interface

**User Interface Description**

The cyclometer is turned on or off with the power switch. The rider starts or stops the ride-time clock with a push of a button and can also reset the clock and distance traveled with a push of the reset button. The rider can change display modes with a simple push of the mode button which takes him/her to the next mode. Figure 1.3 shows what the default state, where the ride time is displayed on the upper line of the 16x2 LCD and the current speed is displayed on line 2 of the LCD.

**Figure 1.3 Mode 1**

16x2 LCD
When the mode button is pushed the cyclometer will cycle to the next mode, which will display
the current ride time on line 1 of the LCD and the total distance traveled on line 2 of the LCD, as
pictured in Figure 1.4.

![Figure 1.4 Mode 2](image)

16x2 LCD

And with an additional push of the mode button the cyclometer will cycle back to mode 2 where
the LCD will display the default mode again, or if time permits another mode will be added in
which the total calories burned will be displayed on line 2 of the LCD which might look like
figure 1.5.

![Figure 1.5 Mode 3](image)

16x2 LCD

Figure 1.6 is a state diagram to help you see how the cyclometer will function.
Figure 1.6

Power

Startup
Turn on LCD
Enable Keys
Enable Interrupts
Prompt user

Default mode

Go/stop

Reset

Mode 1
Ride-time Clock is displayed on Line 1 of LCD
Current Speed is displayed on Line 2

Mode 2
Line 1 displays Clock
Line 2 cleared
Line 2 of LCD displays Distance traveled

Mode 3
Line 1 displays Clock
Line 2 cleared
Line 2 of LCD displays total calories burned

Go/stop

Freezes current display status

Mode

Go/stop

Go/stop

Go/stop
Development Plan

During Fall quarter I pondered different project ideas and narrowed them down to the Cyclometer also, I decided what options my Cyclometer will have. I have decided on my choice of Microcontroller and the programming language I’m going to use. Also I have determined which LCD to use, and have come up with a rough idea of how the appearance of my final design may look, and what dimensions it will make up. Key names and options have also been selected. Recently I have been working on my state diagram, and searching on-line for the best parts for my project. Here is a planned schedule of what work is to be done in the weeks to come.

Fall Quarter 2002

Week 10: Order Magnets, and Hall-effect IC, and buy AA batteries.

Week 11: Finals week, concentrate on Finals.

Xmas Break

Receive parts hopefully; look into the best wire for connecting Sensor to Cyclometer and best way of mounting sensor to bike frame.

Winter Quarter 2003

Week 1: Test Hall-effect sensor with magnet to make sure everything is working correctly.

Week 2: Work on assembling sensor to bike and test to see if harness will hold up to riding vibration.

Week 3: Start writing code, establish basic modes and switching between them.

Week 4: Look into designing my own keypad using 4 individual keys.
Week 5: Set up keypad and LCD and test to see if mode switching works correctly.

Week 6: Set up reset and power buttons

Week 7: Write code for reset and startup and test reset and power keys.

Week 8: Start writing code for Mode 1 and design the Ride-time clock.

Week 9: Finish Mode 1, and start on Mode 2, design algorithm to keep track of total distance traveled.

Week 10: Dead week, concentrate on finals.

Week 11: Finals week, take finals.

Spring Quarter 2003

Week 1: Design algorithm to keep track of current speed.

Week 2: Finish up on algorithms and test them to see if they are working.

Week 3: Start working on power supply for Cyclometer.

Week 4: Finish power supply and look into building circuit board with HC12 and required ports and LCD.

Week 5: Start building Housing for final project and harness to attach cyclometer to bike.

Week 6: Put everything together and begin the test phase.

Week 7: Run tests on distance and speed for accuracy, fix accordingly.

Week 8: If everything is working well, decide on making Mode 3 for calorie burn.

Week 9: Add mode 3 into code, and set up for mode switching.

Week 10: Design Algorithm for calorie burn, and finish code.

Week 11: Fine tune project for demonstration.
**Electrical Specifications:**

**6 Volt Supply:**

<table>
<thead>
<tr>
<th>Component</th>
<th>Current (mA)</th>
<th>Power (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCD</td>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>MCU</td>
<td>45</td>
<td>225</td>
</tr>
<tr>
<td>Hall-effect IC</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td><strong>6.0V source Power Dissipation</strong></td>
<td><strong>68</strong></td>
<td><strong>340</strong></td>
</tr>
</tbody>
</table>

**Accuracy:**

Position: .01 mile  
Velocity: .01 mph  
Time: 1 second

**Power Requirements:**

4 – AA Alkaline Batteries  
Estimated Life 18 hours  
Worst case Power dissipation 340mW

**Special Environment Requirement:**

Water Resistant  
Shock resistant housing  
Shock resistant sensor harness

**Temperature Range**

-5 deg. C to +50 deg. C
## Preliminary Parts List

<table>
<thead>
<tr>
<th>Description</th>
<th>Part #</th>
<th>Sources</th>
<th>Price</th>
<th>Current (max)</th>
<th>Lead Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX6314 reset circuitry</td>
<td>MAX6314US4 S101-T-ND</td>
<td>Digi-Key</td>
<td>$1.04</td>
<td>12uA</td>
<td>2 weeks</td>
</tr>
<tr>
<td>Neodymium magnet</td>
<td>MAG-74</td>
<td>All-electronics</td>
<td>$.50 x2</td>
<td>none</td>
<td>2 days</td>
</tr>
<tr>
<td>LCD 16x2</td>
<td>MDL-16265LV</td>
<td>Digi-Key</td>
<td>$12.82</td>
<td>3 mA</td>
<td>have</td>
</tr>
<tr>
<td>AA Batteries</td>
<td>---------------------</td>
<td>Radio Shack</td>
<td>$.75 x4</td>
<td>--------------</td>
<td>have</td>
</tr>
<tr>
<td>Battery Holder</td>
<td>2478K-ND</td>
<td>Digi-Key</td>
<td>$1.10</td>
<td>-------------</td>
<td>3 weeks</td>
</tr>
<tr>
<td>Microprocessor</td>
<td>MC68HC912B 32CFU8-ND</td>
<td>Digi-Key</td>
<td>$19.26</td>
<td>45 mA</td>
<td>Borrowed 2 weeks</td>
</tr>
<tr>
<td>16MHz Crystal</td>
<td>73-XT49U1600-20</td>
<td>Mouser</td>
<td>.41 x2</td>
<td>-------------</td>
<td>3 weeks</td>
</tr>
<tr>
<td>resistors</td>
<td>varied</td>
<td>Parts room</td>
<td>.08</td>
<td>varies</td>
<td>have</td>
</tr>
<tr>
<td>Push Buttons</td>
<td>JF15SP1H</td>
<td>Allied</td>
<td>2.13 x4</td>
<td>-------------</td>
<td>3 weeks</td>
</tr>
</tbody>
</table>

**TOTAL**  
$50.76  
68mA